Jurnal Ilmiah

Wahana

Volume 1, Nomor 1, Mei 2009 | ISSN: 2085 - 2746 X

Lembaga Penelitian dan Pengabdian Pada Masyarakat (LPPM)
Universitas Baturaja Kabupaten Ogan Komering Ulu Sumatera Selatan

Wahana JI LPPM | Vol. 1 | No. 1 | Hal. 1 - 66 | Baturaja, Mei 2009 | ISSN : 2085 - 2746 XX

Volume 1, Nomor 1, Mei 2009 ISSN: 2085 – 2746 X

Jurnal Ilmiah *Wahana* LPPM Universitas Baturaja

Wahana adalah jurnal ilmiah yang menyajikan berbagai tulisan ilmiah dalam bentuk ringkasan hasil penelitian, artikel ilmiah, dan resensi buku dalam berbagai bidang ilmu, khususnya terkait dengan konsentrasi keilmuan yang diselenggarakan oleh Prodi-Prodi yang ada di lingkungan Universitas Baturaja. Redaksi mengundang para pakar, praktisi, akademisi, peneliti dan siapa saja yang peduli dengan pengembangan ilmu pengetahuan dan teknologi. Diterbitkan secara berkala dua kali dalam satu tahun (Edisi Mei dan November) oleh LPPM Universitas Baturaja Kabupaten Ogan Komering Ulu (OKU) Sumatera Selatan.

Alamat Redaksi:
LPPM Universitas Baturaja
Jl. Ratu Penghulu No. 02301 Karangsari Baturaja OKU Sumsel (32116)
Telepon/Fax.: (0735) 326122– E-mail: LPPM.UNBARA@gmail.com
Contact Person: 0819 – 681400 (Anis Feblin) 0813 – 73478885 (Dorodjatun)

Penerbit:
LPPM Universitas Baturaja
Kabupaten Ogan Komering Ulu (OKU) Sumatera Selatan

Redaksi menerima naskah berupa artikel ilmiah, ringkasan hasil penelitian dan resensi buku dalam berbagai bidang ilmu yang belum pernah diterbitkan oleh media lain. Naskah dikirim dalam bentuk disket/CD file atau via e-mail ke alamat kantor/E-mail Redaksi Jurnal Wahana LPPM Unbara dengan format seperti tercantum pada halaman cover dalam belakang. Redaksi berhak menyunting naskah tanpa mengubah substansi.

Volume 1, Nomor 1, Mei 2009 ISSN 2085 – 2746X

WAHANA

Jurnal Ilmiah LPPM Universitas Baturaja

DAFTAR ISI

Hal :

Kelompok Etnik di Jawa Timur dalam Era Otonomi Daerah
A. Latief Wiyata ..... 1 – 6

Penggunaan Metode Brainstorming *dalam Pembelajaran Menulis*
Bambang Sulistyo ..... 7 – 14

Analisis Pengaruh Independensi, Keahlian Profesional, Frekuensi Audit Atas Laporan Keuangan Historis dan Pengalaman Kerja Pengawas Intern Terhadap Efektivitas Penerapan Struktur Pengendalian Intern
Sri Nova Rina ..... 15 – 23

Penguatan Ekonomi Rakyat Perdesaan dalam Mengentaskan Kemiskinan Perspektif Agribisnis
Munajat ..... 24 – 35

Enskripsi Data Multi Format dengan Metode Berlapis RC-4, ECC dan Vigenere Chiper
Fiftin Noviyanto ..... 36 – 49

Analisis Kelayakan Pengolahan Hasil Pertanian Menjadi Produk Makanan di Kecamatan Sinar Peninjauan Kab. OKU
Yetty Oktarina ..... 50 – 57

Perkembangan Migrasi di Pulau Sumatera
Yunita Sari ..... 58 – 66

Enskripsi Data Multi Format dengan Metode Berlapis RC-4, ECC dan Vigenere Chiper

Oleh: Fiftin Noviyanto*

Abstract

Data is values asset need to keeping from any attack or threat to change, destroy or essential losess. One way to secure of data by encryption. So plaintext can to be secret. There are so many method to secure of data, classic method, modern method or hybrid method. Anymethod have potential to get attack. So, this research mixed three method to encrypt data and we hope the combined make more difficulty to analysis the chipper. From this research, it can be found that RC4, method, ECC method and Vigenere Cipher methode combined effective to protect data and authentication process without significant addition time because ECC or RC4 methode user whose get the data has private key. That key only know by creator.

***Key words**: Attack, threat, encryption, RC4, ECC, Vigenere Cipher, data*

PENDAHULUAN

Berkembangnya teknologi diiringi juga dengan meningkatnya beberapa ancaman terhadap penggunaan teknologi tersebut. Penerapan dalam berbagai bidang sehingga data yang diolah semakin beragam, diantaranya data yang bersifat rahasia. Oleh karena itu, masalah keamanan dan kerahasiaan merupakan salah satu aspek penting dari suatu pesan, data atau informasi, terkait dengan betapa pentingnya pesan, data atau informasi tersebut masih *authenticity*. Hematnya, pesan, data, atau informasi akan tidak berguna lagi apabila pada saat proses pengiriman informasi itu disadap oleh orang yang tidak berhak.

Keamanan dan kerahasian data pada komputer saat ini menjadi isu yang sangat penting dan terus berkembang. Salah satu cara untuk menjaga keamanan dan kerahasiaan pesan, data atau informasi maka diperlukan enkripsi agar tidak dapat di baca atau di mengerti oleh sembarang orang, kecuali untuk penerima yang berhak. File data teks dapat memiliki format yang beragam, misalnya ekstensi .doc, .txt maupun .rtf.

Ada banyak model dan metode enkripsi, diantaranya adalah enkripsi dengan algoritma *Rivest Code 4* (RC4) yang menggunakan teknik kriptografi asimetri yang menggunakan dua buah kunci berbeda dalam proses enkripsi-dekripsi, ECC yang menggunakan teknik kriptografi asimetri yang menggunakan dua buah kunci berbeda dalam proses enkripsi-dekripsi dan metode *vigenere chiper* yang menggunakan teknik *streaming*.

Setiap metode enkripsi selalu memiliki kekurangan, untuk itu diharapkan dengan penggabungan tiga metode, keamanan data dapat lebih baik. Karena kunci pihak yang tidak berkepentingan harus memecahkan tiga lapis kunci terlebih dahulu, dengan masing-masing kunci memiliki teknik yang berbeda.

TINJAUAN PUSTAKA

* *Dosen PNSD Kopertis Wilayah V Dpk di Jurusan Teknik Informatika UAD Yogyakarta*

Cyber Crime

Secara umum, *cyber crime* dapat didefinisikan sebagai tindakan pidana kriminal yang dilakukan pada teknologi internet (*cyberspace*), baik yang menyerang fasilitas umum dalam *cyberspace* ataupun kepemilikan pribadi. Dapat dispesifikasikan menjadi beberapa istilah, diantaranya adalah *computer related crime*, *computer related to computer network*, *computer abuse*, *computer crime*, *e-crime*, *computer related fraud*, *internet crime*, *IT related crime*, *cyber fraud*, dan lain sebagainya. Istilah-istilah yang menggambarkan *cyber crime* dapat persempit lagi menjadi konsep kejahatan computer (*Computer Crime*) dan kejahatan yang berkorelasi dengan computer (*computer related crime*). *Computer crime* dapat didefinisikan sebagai setiap tindakan yang tidak diperbolehkan dalam hukum atau tindakan bisnis convensional (transaksi) yang membahayakan baik itu terhadap person ataupun property yang menggunakan system teknologi informasi. Sementara itu, *computer related crime* dapat dijabarkan menjadi dua buah kategori utama, yakni computer sebagai target kejahatan serta penggunaan computer sebagai alat untuk melakukan kejahatan yang dijabarkan sebagai berikut;

a. Komputer sebagai target kejahatan, meliputi; Sabotase sistem komputer dan jaringan komputer; Sabotase sistem operasi dan program-program komputer; Pencurian data-data atau informasi; Pencurian properti-properti intelektual, seperti software komputer; Pencurian informasi-informasi marketing, dan; Data-data yang disarikan dari file-file yang terkomputerisasi, seperti data-data medis, daftar riwayat hidup seseorang, data-data finansial dan sebagainya.
b. Komputer sebagai instrumen untuk melakukan kejahatan, meliputi; Penipuan *account* dalam transaksi yang melibatkan *Automatic Teller Machine* (ATM); Penipuan kartu kredit; Penipuan yang meliputi tranfer dana secara elektronik; Penipuan telekomunikasi, dan; Penipuan yang terkait dengan *Electronic Commerce* dan Electronic Data Interchange (EDI).

Kriptografi

Seharusnya keamanan dan kerahasian suatu pesan, data, ataupun informasi adalah merupakan hal yang mutlak yang harus kita lakukan. Sedangkan alat untuk melakukan pengamanan data dalam sistem komunikasi jaringan komputer sering disebut *cryptography*. Kriptografi (*cryptography*) merupakan ilmu dan seni penyimpanan pesan, data, atau informasi secara aman. Kritptanalisis (*cryptanalysis)* merupakan ilmu dan seni pembongkaran pesan, data, atau informasi rahasia seperti di atas.

Kriptologi (*cryptology*) adalah panduan dari kriptografi dan kriptanalist. Fungsi-fungsi yang mendasar dalam kriptografi adalah enkripsi dan dekripsi. Enkripsi adalah proses mengubah suatu pesan, data, atau informasi asli (*plaintext*) menjadi suatu pesan, data, atau informasi dalam bahasa sandi (*ciphertext*). Sedangkan dekripsi adalah proses mengubah pesan, data, atau informasi dalam suatu bahasa sandi kembali menjadi pesan, data, atau informasi asli. Berikut ini adalah hal-hal penting yang dicakup dan sering dibahas dalam teori kriptografi.

Kunci Simetris

Kunci Simetris adalah jenis kriptografi yang paling umum digunakan. Kunci untuk membuat pesan yang di sandikan sama dengan kunci untuk membuka pesan yang disandikan itu. Jadi pembuat pesan dan penerimanya harus memiliki kunci yang sama persis. Siapapun yang memiliki kunci tersebut termasuk pihak-pihak yang tidak diinginkan dapat membuat dan membongkar rahasia *ciphertext*. Contoh algoritma kunci simetris yang terkenal adalah DES (*Data Encryption Standard*).

Kunci Asimetris

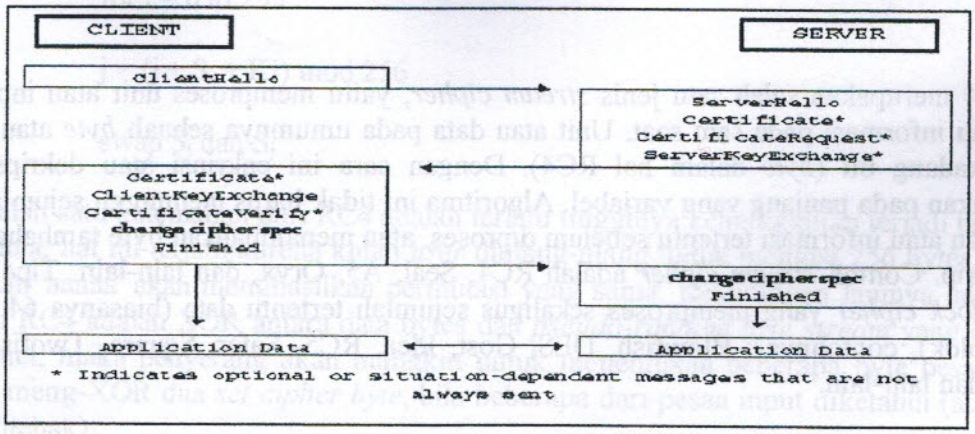
Kunci asimetris adalah pasangan kunci kriptografi yang salah satunya digunakan untuk proses enkripsi dan yang satu lagi untuk dekripsi. Semua orang yang mendapatkan kunci publik dapat menggunakannya untuk mengenkripsikan suatu pesan, data ataupun informasi, sedangkan hanya satu orang saja yang memiliki rahasia tertentu dalam hal ini kunci privat untuk melakukan pembongkaran terhadap sandi yang dikirim untuknya. Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA.

Fungsi Hash Satu Arah

Fungsi *hash* satu arah (*one-way hash function*) digunakan untuk membuat sidik jari (*fingerprint*) dari suatu dokumen atau pesan X. Pesan X (yang besarnya dapat bervariasi) yang akan di-*hash* disebut *pre-image*, sedangkan outputnya yang memiliki ukuran tetap, disebut *hash-value* (nilai *hash*). Fungsi *hash* dapat diketahui oleh siapapun, tak terkecuali, sehingga siapapun dapat memeriksa keutuhan dokumen atau pesan X tersebut. Tak ada algoritma rahasia dan umumnya tak ada pula kunci rahasia. Contoh algoritma fungsi *hash* satu arah adalah MD-5 dan SHA. *Message Authentication Code* (MAC) adalah salah satu variasi dari fungsi *hash* satu arah, hanya saja selain *pre-image*, sebuah kunci rahasia juga menjadi input bagi fungsi MAC.

Secure Socket Layer (SSL)

SSL dapat menjaga kerahasiaan (*confidentiality*) dari informasi yang dikirim karena menggunakan teknologi enkripsi yang maju dan dapat di-*update* jika ada teknologi baru yang lebih bagus. Dengan penggunaan sertifikat digital, SSL menyediakan otentikasi yang transparan antara *client* dengan *server*. SSL menggunakan algoritma RSA untuk membuat tanda tangan digital (*digital signature*) dan amplop digital (*digital envelope*) (http://bdg.centrin.nrt.id/~budskman/protek.htm). Selain itu, untuk melakukan enkripsi dan dekripsi data setelah koneksi dilakukan, SSL menggunakan RC4 sebagai algoritma standar untuk enkripsi kunci simetri. Saat aplikasi menggunakan SSL, sebenarnya terjadi dua kondisi, yakni *handshake* dan pertukaran informasi.



Biasanya, *browser-browser* seperti *Netscape Navigator* atau *Microsoft Internet Explorer* sudah menyertakan sertifikat digital dari CA utama yang terkenal, sehingga memudahkan pemeriksaan sertifikat digital pada koneksi SSL. Penyertaan serfikat digital CA utama pada *browser* akan menghindarkan *client* dari pemalsuan sertifikat CA utama.

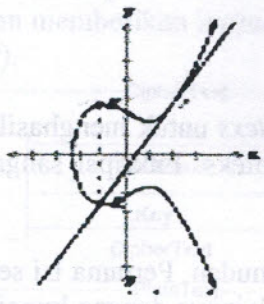
Serangan Penukaran Pesan Melalui Jaringan Komputer

Berdasarkan bagaimana cara dan posisi seseorang mendapatkan pesan-pesan dalam saluran komunikasi, penyerangan dapat dikategorikan menjadi: *Sniffing*, *Replay attack*, *Spoofing* dan *Man-in-the-middle*.

METODE PENELITIAN

Eliptic Curve Cryptography (ECC)

ECC merupakan teknik kriptografi asimetri yang menggunakan dua buah kunci berbeda dalam proses enkripsi-dekripsi, dikenal dengan *private key*, dan *public key*. Persamaan kurva elips ternomalisasi yang digunakan adalah; $y^3 \equiv x^3 + -6x + 6 (\text{mod} 257)$. Pemilihan mod 257 dikarenakan pada sistem ini, metoda pengacakan data dilakukan tiap *byte* dalam suatu koordinat tertentu. Angka maksimal satu *byte* adalah 255, maka bilangan prima terdekat dengan 255 adalah 257.



Gambar 1. Kurva elips untuk persamaan $y^2 = x^3 - 6x + 6$

RC4

RC4 merupakan salah satu jenis *stream cipher*, yaitu memproses unit atau input data, pesan atau informasi pada satu saat. Unit atau data pada umumnya sebuah *byte* atau bahkan kadang kadang bit (*byte* dalam hal RC4). Dengan cara ini enkripsi atau dekripsi dapat dilaksanakan pada panjang yang variabel. Algoritma ini tidak harus menunggu sejumlah input data, pesan atau informasi tertentu sebelum diproses, atau menambahkan byte tambahan untuk mengenkrip. Contoh *stream cipher* adalah RC4, Seal, A5, Oryx, dan lain-lain. Tipe lainnya adalah *block cipher* yang memproses sekaligus sejumlah tertentu data (biasanya 64 bit atau 128 bit blok), contohnya : Blowfish, DES, Gost, Idea, RC5, Safer, Square, Twofish, RC6, Loki97, dan lain-lain.

RC4 merupakan enkripsi *stream simetrik proprietary* yang dibuat oleh *RSA Data Security Inc* (RSADSI). Penyebarannya diawali dari sebuah *source code* yang diyakini sebagai RC4 dan dipublikasikan secara *'anonymously'* pada tahun 1994. Algoritma yang dipublikasikan ini sangat identik dengan implementasi RC4 pada produk resmi. RC4 digunakan secara luas pada beberapa aplikasi dan umumnya dinyatakan sangat aman. Sampai saat ini diketahui tidak ada yang dapat memecahkan/membongkarnya, hanya saja versi ekspor 40 bitnya dapat dibongkar dengan cara "*brute force*" (mencoba semua kunci yang mungkin). RC4 tidak dipatenkan oleh RSADSI, hanya saja tidak diperdagangkan secara bebas (*trade secret*).

Algoritma RC4 cukup mudah untuk dijelaskan. RC4 mempunyai sebuah *S-Box*, $S_0,S_1,...,S_{255}$, yang berisi permutasi dari bilangan 0 sampai 255, dan permutasi merupakan fungsi dari kunci dengan panjang yang variabel. Terdapat dua indeks yaitu i dan j, yang diinisialisasi dengan bilangan nol. Untuk menghasilkan random byte langkahnya adalah sebagai berikut :

$$i = (i + 1) \bmod 256$$

$$j = (j + S_i) \bmod 256$$

swap S_i dan S_j

$$t = (S_i + S_j) \bmod 256$$

$$K = S_t$$

Byte K di XOR dengan *plaintexs* untuk menghasilkan *cipherteks* atau di XOR dengan cipherteks untuk menghasilkan plainteks. Enkripsi sangat cepat kurang lebih 10 kali lebih cepat dari DES.

Inisialisasi S-Box juga sangat mudah. Pertama isi secara berurutan $S_0 = 0$, $S_1 = 1,...,S_{255} = 255$. Kemudian isi array 256 byte lainnya dengan kunci yang diulangi sampai seluruh array $K_0, K_1,...,K_{255}$ terisi seluruhnya. Set indeks j dengan nol, Kemudian lakukan langkah berikut :

for i = 0 to 255

$j = (j + S_i + Ki) \bmod 256$

swap S_i dan S_j

Salah satu kelemahan dari RC4 adalah terlalu tingginya kemungkinan terjadi tabel S-box yang sama, hal ini terjadi karena kunci *user* diulang-ulang untuk mengisi 256 bytes, sehingga 'aaaa' dan 'aaaaa' akan menghasilkan permutasi yang sama. Kekurangan lainnya ialah karena enkripsi RC4 adalah XOR antara data bytes dan *pseudo-random byte stream* yang dihasilkan dari kunci, maka penyerang akan mungkin untuk menentukan beberapa byte pesan orisinal dengan meng-XOR dua *set cipher byte*, bila beberapa dari pesan input diketahui (atau mudah untuk ditebak).

Vigenere Chipper

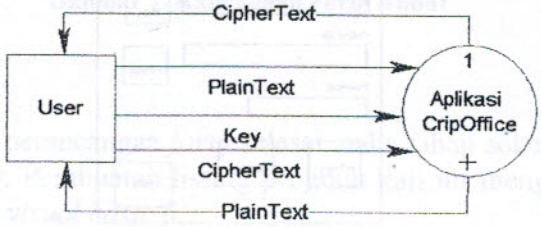
Metode *Vigenere Chiper* merupakan metode untuk mengenkripsi data alphabet. Metode ini varian dari *Caesar Chiper* yang hanya menggunakan satu kunci. Untuk mengenkripsi data, dengan algoritma substitusi abjad majemuk. Metode ini memiliki kelebihan pada keamanan terhadap frekuensi huruf, karena setiap huruf yang sama pada plain teks tidak selalu sama pada chiper teks. Hal tersebut tergantung dari kunci yang digunakan. Algoritma enkripsi metode *vigenere cheaper* sebagai berikut : $C = (P + K) \bmod 26$. Sedangkan algoritma untuk deskripsinya adalah : $P = (C - K) \bmod 26$, di mana: C = Chiper teks, P = Plain teks dan K = Kunci

HASIL DAN PEMBAHASAN

Pemodelan sistem

a. Diagram konteks

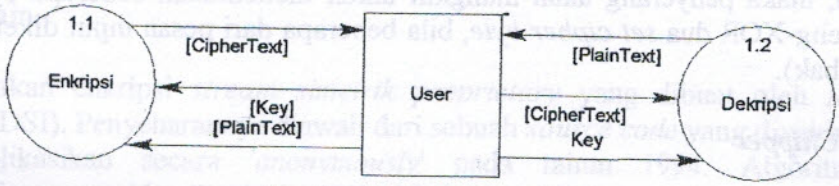
Diagram konteks merupakan penggambaran dari seluruh sistem yang akan dibuat. Tahap ini akan memperjelas hubungan antara *input* yang diberikan oleh *user* ke sistem dan *output* dari sistem kepada *user*. Dalam perancangan sistem ini akan terdapat sebuah entitas yaitu user yang akan memberikan masukan ke sistem yang berupa *file* atau dokumen yang berekstensi .doc, .txt, .rtf, .xls, dan .jpeg dalam bentuk *plaintext* ataupun *ciphertext*. Dari hasil masukan yang diberikan oleh *user*, sistem akan memberikan *output* berupa dokumen dari hasil enkripsi (*ciphertext*) atau dekripsi (*ciphertext*).



Gambar 2. Diagram Konteks

b. **Perancangan DFD (*Data Flow Diagram*)**

Setelah merancang sebuah diagram konteks, kemudian akan dirancang DFD yang akan menjadi penggambaran dari seluruh sistem yang akan dibuat. DFD sebuah cara untuk memodelkan alur data yang terjadi pada sebuah sistem. Dari diagram konteks yang telah kemudian akan dijelaskan dengan penggambaran sebuah diagram alir data. Diagram ini akan menjelaskan bagaimana proses yang terjadi sebelum sebuah *file* atau dokumen yang dimasukkan oleh user menjadi *output* yang berupa *file* atau dokumen dari hasil enkripsi atau dekripsi. Lihat gambar 3.



Gambar 3. DFD Level 0

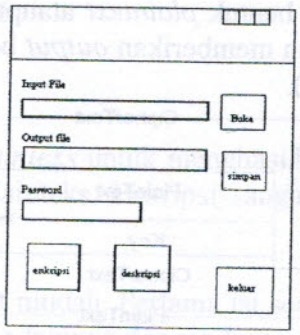
Alur data yang terjadi pada penggambaran DFD level 0 sebagai berikut:

1) *User* memasukkan *plaintext* atau dokumen dan *key*, kemudian pada proses 1.1 hasil masukan *file* yang diberikan oleh *user* akan dienkripsi. Proses enkripsi akan menghasilkan *output file* enkrip berupa *ciphertext* kepada *user*.
2) Apabila *user* akan medekripsi *file* maka data dari *user* akan masuk ke proses 1.2 yang berupa *ciphertext* dan *key* yang akan didekripsi. Setelah *file* dan *key* didekripsi maka akan dihasilkan *output file* dekrip berupa *plaintext* kepada *user*.

c. **Perancangan *Input-Output***

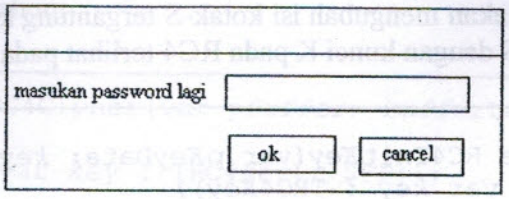
Tahap perancangan *input-output* merupakan tahap perancangan *interface*, agar program aplikasi dapat berinteraksi dangan baik dengan *user*. Sehingga *output* yang dikeluarkan sesuai dengan harapan yang diinginkan oleh *user*. Setelah pembuatan algoritma diatas akan dirancang lima *form*, yaitu *form* utama, *form password* dialog, *form* step dan *form about*.

Rancangan *form* (lihat gambar 4) akan digunakan untuk membuat *form* utama yang akan berfungsi sebagai tempat untuk memproses file atau dokumen yang akan dienkripsi dan dekripsi.



Gambar 4. Rancangan Form Utama

Rancangan *form* dibawah akan digunakan untuk membuat *form password* dialog yang berfungsi untuk menulis ulang *password* sehingga *password* yang dimasukkan adalah benar.

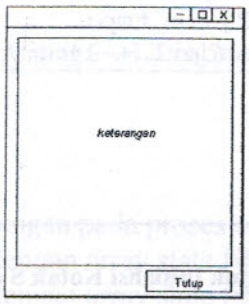


Gambar 5. Rancangan Form Verify *Password*

Rancangan *form* berikut akan digunakan untuk membuat *formhelp* yang berisi langkah atau cara menggunakan program yaitu langkah mengenkripsi dan mendekripsi file.

Gambar 6. *Rancangan Form Step*

Rancangan *form* dibawah akan digunakan untuk membuat *form About* yang berisi informasi identitas pembuat program atau programmer.



Gambar 7. Rancangan Form About

Implementasi Sistem

Setelah algoritma dan perancangan *form* selesai maka tahap selanjutnya adalah membuat listing program atau *coding*. Pembuatan listing program kali ini menggunakan bahasa Pascal dan menggunakan *compiler visual basic 6*.

a. Inisialisasi Kotak S dan mengubah Kotak S dengan kunci K pada RC4

RC4 mempunyai sebuah S-Box, S0,S1,.....,S255, yang berisi permutasi dari bilangan 0 sampai 255. Dan RC4 akan mengubah isi kotak-S tergantung kunci K. Inisialisasi Kotak S dan mengubah Kotak S dengan kunci K pada RC4 terlihat pada listing di bawah ini.

```
[1] procedure RC4InitKey(var pKeyData; keyDataLen :
    integer; var key : TRC4Key);
[2] var
[3]   index1,index2 : Integer;
[4]   counter : Integer;
[5]   keyData : array [0..0] of Byte absolute pKeyData;
[6]   tmp : Integer;
[7] begin
[8]   with key do
[9]   begin
[10]    for counter := 0 to 255 do
[11]      state[counter] := counter and $FF;
[12]    i := 0;
[13]    j := 0;
[14]    index1 := 0;
[15]    index2 := 0;
[16]    for counter := 0 to 255 do
[17]    begin
[18]      index2 := ((keyData[index1] +state[counter]
              + index2)) and $FF;
[19]     { SWAP }
[20]     tmp := state[counter];
[21]     state[counter] := state[index2];
[22]     state[index2] := tmp;
[23]     index1 := (index1 + 1) mod keyDataLen;
[24]    end;
[25]  end;
[26]end;
```

Listing ---. Listing Untuk Inisialisi Kotak S dan Mengubah dengan kunci K

Keterangan listing :

1) Baris 11 – 12 adalah listing untuk inisialisasi kotak S pada array berukuran 256 byte. Array 256 byte diidentifikasikan oleh array state yaitu array [0..255].
2) Baris 17 – 25 adalah proses random byte dimana SBox bernilai random terhadap KBox atau mengubah isi Sbox dengan kunci K.
3) Pada listing diatas terdapat $FF yaitu bentuk hexadecimal dari decimal 256.

b. Metode RC4

Untuk membangkitkan kunci enkripsi atau dekripsi dengan metode RC4 menggunakan listing program dibawah ini.

```
  procedure RC4Cipher(var pBuffer; bufferLen :
     integer;var key : TRC4Key); $FF]);
[1] var
[2]   si, sj : Integer;
[3]   counter : Integer;
[4]   buffer : array [0..0] of Byte absolute pBuffer;
[5]
[6] begin
[7]   with key do
[8]   begin
[9]     for counter := 0 to bufferLen-1 do
[10]    begin
[11]      i := (i + 1) and $FF;
[12]      si := state[i];
[13]      j := (j + si) and $FF;
[14]      sj := state[j];
[15]    { SWAP }
[16]      state[j] := si;
[17]      state[i] :=sj;
[18]    buffer[counter] := Byte(buffer[counter] xor
                           state[(si + sj) and $FF]);
[19]    end;
[20]  end;
[21]end;
```

Listing ---. Listing Metode RC4

Keterangan listing :

1) Baris 10 – 19 adalah proses perulangan pada proses enkripsi atau dekripsi.
2) Baris 13 adalah inisialisasi isi si dengan array state [i] pada operasi baris 12.
3) Baris 15 adalah inisialisasi isi sj dengan array state [j] pada operasi baris 14.
4) Baris 17 dan 18 adalah proses pertukaran isi Sbox index ke-i dengan isi Sbox index ke-j.
5) Baris 19 adalah proses XOR pada data *buffer* dengan isi Sbox. Pada proses ini lakukan perulangan sampai data pada *buffer* habis.

c. Metode ECC

Untuk membangkitkan enkripsi dengan metode ECC menggunakan listing program di bawah ini.

```
[1]  function ECC(var x:integer;);

[2]  var
[3]    enkripecc : Integer;

[4]  begin
[5]        enkripecc=sqrt(sqrt((x*x*x+(-6)*x+ 6)mod 257) )
[6]  end;
```

Listing ---. Listing Metode ECC

Keterangan listing :

1) Baris 2 – 3 adalah proses inisialisasi variable enkripsiecc.
2) Baris 5 adalah proses enkripsi mengunakan metode ECC. Nilai x yang diproses merupakan hasil proses enkripsi menggunakan metode RC4. Hasil proses enkripsi tersebut disimpan dalam variable enkripsiecc.

d. Metode Vigenere Cipher

```
[1] function vigenere(var p,k:integer;);

[2] var
[3]   enkripvig : Integer;

[4] begin
[5]       enkripvig= (p+k) mod 26
[6] end;
```

Keterangan listing :

1. Baris 2 – 3 adalah proses inisialisasi variable enkripsivig.
2. Baris 5 adalah proses enkripsi mengunakan metode Vigenere. Nilai p dan k diperoleh dari hasil proses enkripsi menggunakan metode ECC. Hasil proses enkripsi tersebut disimpan dalam variable enkripsivig.

e. Metode penggabungan RC4, ECC dan Vigenere Cipher

```
procedure RC4Cipher(var pBuffer; bufferLen :
    integer;var key : TRC4Key); $FF]);
[1] var
[2]   encripecc : array [0..n] of integer absolute pBuffer;
[3]   si, sj : Integer;
[4]   counter : Integer;
[5]   buffer : array [0..n] of Byte absolute pBuffer;
[6]
[7] begin
```

```
[8]   with key do
[9]   begin
[10]      for counter := 0 to bufferLen-1 do
[11]      begin
[12]        i := (i + 1) and $FF;
[13]        si := state[i];
[14]        j := (j + si) and $FF;
[15]        sj := state[j];
[16]       { SWAP }
[17]          state[j] := si;
[18]          state[i] :=sj;
[19]      buffer[counter] := Byte(buffer[counter] xor
                             state[(si + sj) and $FF]);
[20]
[21] for counter := 0 to bufferLen-1 do
[22] enkripecc[counter]=sqrt(sqrt((x*x*x+(-6)*x+
                                  6)mod 257) );
[23] vigenere(enkripecc[counter],key);

        end;
     end;
  end;
```

Listing ---. Listing Gabungan Metode RC4 dan ECC

Keterangan listing :

1) Baris 1 – 6 adalah proses inisialisasi variable yang akan digunakan pada proses enkripsi.
2) Baris 7 - 19 adalah proses enkripsi mengunakan metode RC4 dan hasil enkripsi merupakan suatu array yang disimpan dalam variable *buffer*[counter].
3) Baris 21-22 adalah memproses hasil enkripsi RC4 yang berupa array, dilanjutkan dengan proses ECC dan disimpan dalam variable enkripsiecc[counter].
4) Baris 23 adalah memproses hasil enkripsi ECC kemudian dilanjutkan dengan metode Vigenere Cipher dengan pemanggilan fungsi.

f. Komputasi Kunci

```
[1]   procedure encriptcr;
[2]   var
[3]   i, j, l, tem, t, y : Integer;
[4]   kEncript : byte;
[5]   s[]:array[1..256];
[6]   k[]:array[1..256];
[7]
[8]   begin
[9]      j := 0;
[10]     l = Len(s);
[11]     For i = 0 To 255
[12]       begin
[13]           j := i Mod l + 1;
[14]           k(i) := Asc(Mid(s, j, 1));
[15]           i++;
[16]       end;
[17]
```

```
[18]        For i = 0 To 255
[19]           Begin
[20]              j := (j + s(i) + k(i)) Mod 256
[21]              Tem: = s(i)
[22]              s(i) := s(j)
[23]              s(j) := s(i)
[24]             t := (s(i) + s(j)) Mod 256
[25]             y := t
[26]             kEncrypt(i) := y
[27]             i++;
[28]          End;
[29]    End.
```

Listing ---. Listing Pembuatan Kunci

Keterangan listing :

1) Baris 2 – 6 adalah proses inisialisasi variable yang akan digunakan pada proses enkripsi.
2) Baris 9 - 16 adalah proses membaca file yang dienkripsi dari lokasi file yang diinputkan, selanjutnya file tersebut dibuat menjadi biner.
3) Baris 18 - 28 adalah proses menukarkan nilai dari file yang telah diubah menjadi biner dengan kunci yang digunakan. Sehingga file asli menjadi file terenkripsi dengan kunci yang telah ditentukan.

KESIMPULAN

1. Data yang memiliki nilai yang bersifat strategis mutlak membutuhkan pengamanan, terlebih pada data yang melewati jalur internet.
2. Tidak ada satupun system yang dapat menjamin kemanan data secara mutlak, namun dengan implementasi enkripsi data dapat membantu melindungi data pada tingkatan tertentu.
3. Penggabungan beberapa metode enkripsi dapat memberikan keamanan berlapis sehingga akan mempersulit kriptoanalisis melakukan attack terhadap data.

DAFTAR PUSTAKA

Ariyus, Dony. 2008. *Pengantar Ilmu Kriptografi*. Yogyakarta: Andi Offset.

Caroll, John M. 1989. *Handbook of Lost Provention an Crime Prevention 2nd edition*". Butterworth-Heinemann.

Ismail, Kais dan Eko Aribowo Eko. 2002. *Organisasi Berkas Teknik Informatika Universitas Ahmad Dahlan*. Yogyakarta: Universitas Ahmad Dahlan.

Kristianto, Andi. 2003. *Keamanan Data Pada Jaringan Komputer*. Yogyakarta: Gava Media

Kumar, Atul. 2002. *Cyber Crime: Crime Without Punishment*".

Meliala, A. E. dan Widags K. 2005. "Etiologi Cybercrime", Makalah disampaikan dalam Seminar Nasional dan Workshop Information Technology (IT) Security, 19 Maret 2005.

Wahana Komputer Semarang. 2003. *Memahami Model Enkripsi dan Security Data.* Yogyakarta: Andi Offset.

Internet:

http://bdg.centrin.nrt.id/~budskman/protek.htm

www.stttelkom.ac.id/staf/UKU/Publikasi%20Ilmiah%20UKE/Jurnal%20E**ECC**IS2006%20(UNBRAW).doc

http://trumpetpower.com/papers/crypto/playfair

http://mohtar.staff.uns.ac.id/files/2009/03/cybercrime.pdf